

Creating Temporal Abstractions in Three Clinical Information Systems

Michael G. Kahn, M.D., Ph.D. and Keith A. Marrs M.S.

Section of Medical Informatics, Washington University, St. Louis, Missouri

Modern clinical information system developers recognize the need to associate temporal information with clinical data. However, specific clinical systems capture different temporal features using a variety of data modeling techniques. Two commonly used methods to represent temporal information are point-based events and interval-based durations. We recently implemented a rule-based expert system for drug dose monitoring on three clinical information systems. The expert system requires both static drug dosing information (drug name, amount, route, frequency) and temporal dosing information (duration of therapy, renewals, restarts). Our design goal was to use the same expert system code on all three information systems by defining a common database schema to hide differences in the original systems' data models. Although we have been successful in mapping clinical data from these three source systems into a unified temporal data representation, we describe how differences in handling time within the three clinical systems made this goal difficult to achieve.

INTRODUCTION

Starting with the original time-oriented data (TOD) model, time has been recognized as being an especially critical component of clinical information.[1] The database and medical informatics communities both have active on-going research activities in modeling, representing, and querying temporal information in electronic databases.[2-5] The temporal database community has defined two key temporal dimensions: *valid time* is the time when a fact is true in the modeled reality, *transaction time* is the time when a fact is current in a database.[6] Although new temporal data models have appeared in research prototypes,[7, 8] most commercial clinical information systems (CISs) use simple temporal representations. The two most common representations are point-based events and interval-based durations.

Point-based event representations associate a single time-stamp, usually representing the valid time with each new data element. Interval-based representations associate two time-stamps, a start-time and stop-time, representing the valid duration of the tuple. In both representations, a second "system" time-stamp frequently is included, representing the transaction time

of the tuple. These two representations are not exclusive; most current clinical information systems contain both event- and duration-based temporal concepts.

We have developed an expert system for drug dose monitoring for three clinical information systems at Barnes and Jewish Hospitals: Systematics (formerly TDS) HC-7000, Emtex System 2000, and a COBOL-based self-developed pharmacy system. To prevent the need to maintain different versions of the expert system, we wished to create a single unified temporal data model for drug orders irrespective of the source system's data model. We describe how differences in temporal representations in these three systems required significant processing to achieve a unified temporal representation.

REPRESENTING DRUG DOSING

DoseChecker is a rule-based expert system which identifies patients who may be receiving either excessive or insufficient drug dosing based on patient demographics, renal function, and other relevant patient findings.[9] DoseChecker requires static drug dosing information (drug name, amount, route, frequency) and temporal dosing information (duration of therapy, renewals, restarts). We defined a unified interval-based temporal representation of the drug dosing information which could be used by DoseChecker (Table 1). Our goal was to create the required views from the three source databases into the same representation so that unmodified expert system code could be used in any system.

Figure 1 illustrates the set of possible temporal representations of medication orders and drug dosing which could be found in a clinical information system. *Medication orders* denote when a medication order was initiated, terminated, held, or renewed. *Medication Administration Events* denotes when a patient actually receives a medication. *Order intervals* denotes an interval of time during which an order for a specific drug, amount, route, and frequency was active. *Drug Intervals* record the time during which a patient received a specific drug at a specific amount, route, and frequency. Drug Intervals can combine multiple order intervals under specific conditions (see

Table 1: Unified Representation of Drug Dosing Information

Attribute Name	Meaning
Registration Number	unique patient identifier
Order Number	unique order identifier
Expanded Order Number	see text
Start Date	start of drug order
Stop Date	conclusion of drug order
Drug Class	NDC drug class of order
Drug Code	NDC drug code of order
Amount	numeric drug amount
Units	units of drug amount
Route	administration route
Frequency	drug dosing frequency

below). *Drug courses* record the time during which a patient received a specific drug, even if the amount, frequency or route changed. Figure 1 illustrates these distinctions for a patient receiving a tapering course of intravenous and oral Prednisone. The unified schema shown in Table 1 is used to model both order- and drug-interval abstractions. To perform the required reasoning tasks, DoseChecker requires order- and drug-interval abstractions to be generated from the source CIS systems.

METHODS

Data from the Barnes CIS and Systematics (TDS) systems are generated from custom queries which run as nightly batch jobs. Text files created from these queries are transferred to a Sun Microsystems UNIX server, are processed by text processing scripts and are loaded into a Sybase relational database (RDBMS).[10] Further processing using SQL scripts are required to create tuples with the semantics and format required for Table 1. Thus, the processing required to normalize tuples occurs in three locations: in mainframe-based queries, in UNIX-based text scripts, and in RDBMS-based SQL scripts.[11]

Data from EMTEK are transferred to a Sybase relational database on a Sun Microsystems UNIX server using the DataBridge product provided by the vendor. Data transfers from EMTEK into Sybase occur every two hours. Data are transformed into the format required for Table 1 by defining a relational view over multiple Sybase tables.

RESULTS

We describe our results using the five classes of temporal abstractions shown in Figure 1.

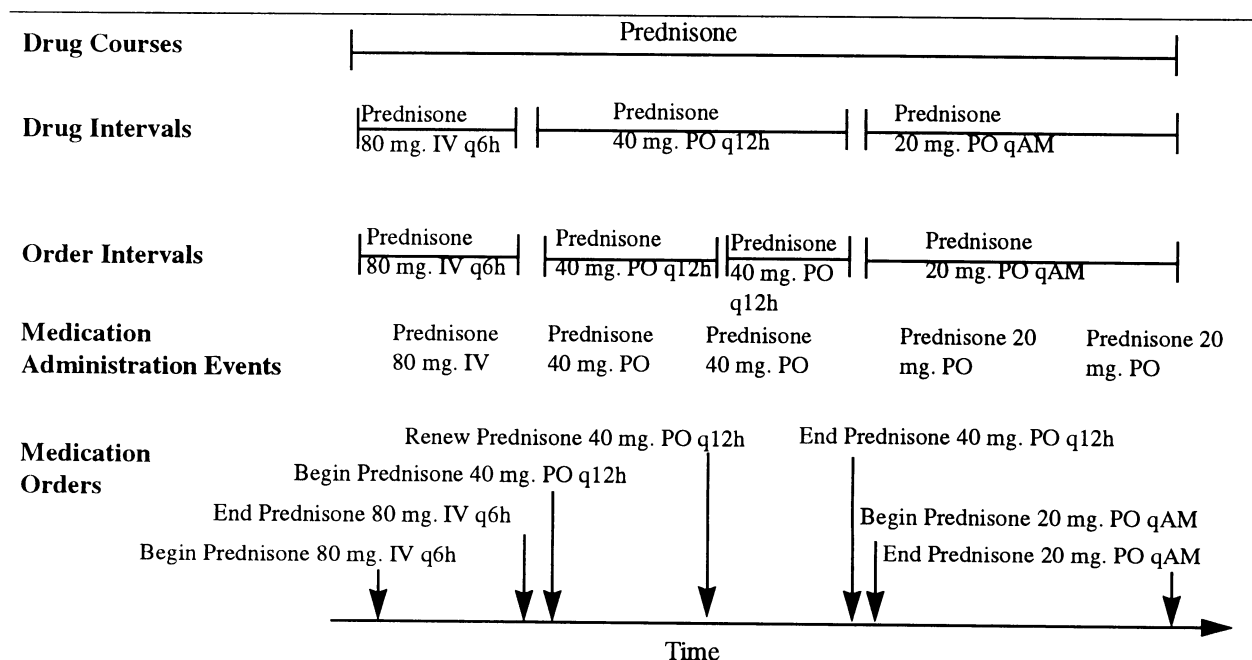


Figure 1: Temporal Abstractions

Emtek and Systematics can generate tuples which denote individual drug administration events. However, we did not use these tuples to generate the required abstractions directly because:

- it was too difficult to infer dosing frequencies reliably from absolute administration times
- more than one medication administration event must occur before a dosing frequency can be inferred. To be most effective, DoseChecker requires a dosing frequency as early as possible.

All three systems record medication orders. Medication orders include a drug name, amount, units, and frequency. The three systems use different techniques to distinguish among medication start, stop, hold, or renew orders. The Barnes and EMTEK system maintain a unique order identifier which can be used to link a medication stop, hold, or renew order to a specific medication start order. Although Systematics records start, stop, hold, and renew orders, the system assigns each of these orders a unique order key. Thus in the Systematics database, there is no direct way to determine to which start order a specific stop order refers. This missing linkage causes significant difficulties in creating order- and drug-interval abstractions.

The presence of a medication start order is sufficient information to create an initial order interval. This tuple, with a null-valued stop time, is considered “cur-

In EMTEK, a database view can be created directly from the primary source data to create an order-interval view. However, because the data model used by EMTEK’s Data Bridge product is not a “clinical” model, a complex three-table equi-join using non-intuitive attributes is required (Figure 2). Despite this complex join, the EMTEK system requires the least extraneous processing to achieve the desired order interval abstraction.

Drug intervals represent the period of time a patient received a specific drug at a uniform amount, route, and frequency, irrespective of the number of orders written during the course of therapy. Drug interval abstractions are created by the temporal catenation of temporally adjacent intervals when specific conditions hold. For this task, two intervals are considered tem-

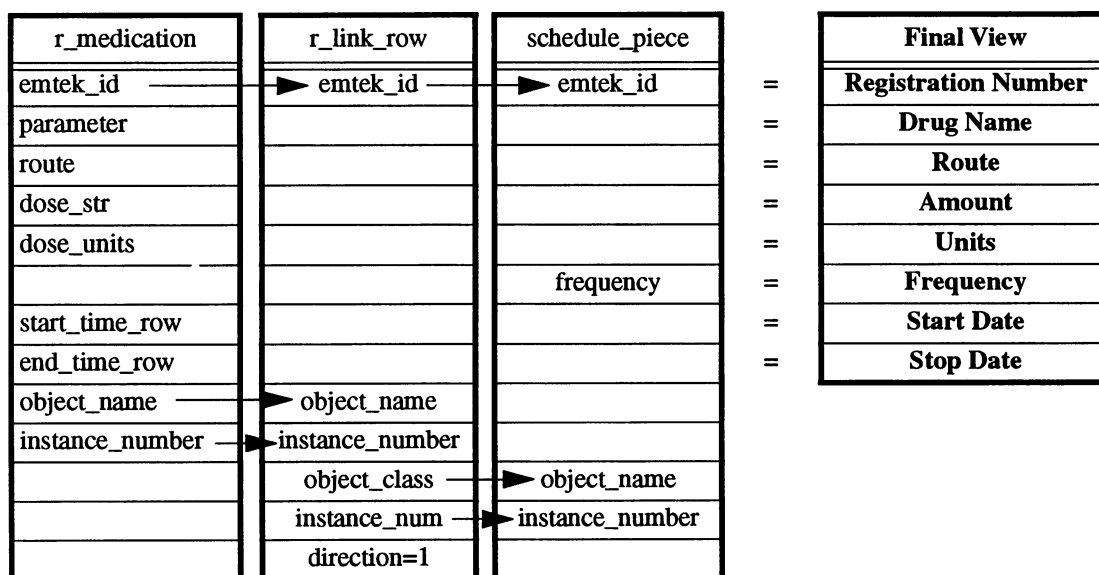


Figure 2: EMTEK three-table join for order interval abstraction view. Arrows denote equi-join attributes.

porally adjacent if the absolute difference between the stop time of the first interval is within 24 hours of the start time of the second order interval. In addition to temporal adjacency, the two intervals must represent the same drug, amount, route, and frequency. Two order intervals can be joined to form a drug interval. In addition, a drug interval can be extended by joining the original drug interval with a temporally adjacent order interval.

We use the same table to store both order intervals and drug intervals. Tuples with positive order numbers represent order intervals; tuples with negative order number represent drug intervals. Order intervals which are contained within a drug interval abstraction contain the drug interval order number in the expanded order number attribute. Order intervals which do not participate in a drug interval have a NULL value in the expanded order number attribute. We have defined two views on this table:

- an “orders” view removes all drug interval abstractions
- a “drugs” view combines drug intervals and singleton (noncatenated) order intervals

The “orders” view captures the sequence of drug orders as received from the pharmacy system. The “drugs” view captures the clinical sequence of drug dosing.

None of the source systems provides drug interval abstractions directly. Creating new drug intervals or extending existing drug intervals is performed using a set of complex SQL scripts. A key difficulty is describing the temporal constraints between two candidate intervals in SQL.

Additional conditions ensure that the drug name, amount, frequency, and route are identical. If the two intervals are order intervals, a new drug interval must be inserted; if the first interval is a drug interval, its stop date must be extended to the stop date of the second interval.

Drug Courses

Drug courses are a further generalization of drug interval abstractions. Drug courses represent the period of time a patient received a specific drug irrespective of changes in the amount, route or frequency of administration.

As with drug intervals, none of the source systems produce drug course abstractions. We generate drug courses from tuples in Table 1 using the same defini-

tion of “temporally adjacent” but removing the constraint that the drug amount, route, and frequency must be identical for merging. Although the DoseChecker expert system has not required drug courses, a number of drug utilization studies have used this abstraction to obtain the length of drug treatment for drugs targeted for pharmacy cost-control measures.[12]

DISCUSSION

The three clinical systems described here each have temporal information associated with pharmacy-related data. The diversity of primary data elements and temporal representations has required significant additional processing to populate a unified global temporal representation of drug dosing intervals. We do not use all available dosing information; for the DoseChecker task, medication administration events were not relevant and therefore have not been stored. These point events would be mandatory if we expand our expert system reasoning task to include the interpretation of peak and trough drug level laboratory results.

The RESUME system, developed by Shahar, employs a variety of techniques to describe, detect, and create temporal abstractions.[13] The RESUME system could easily represent the temporal adjacency and equi-join conditions to create drug intervals and drug courses from order intervals. The example in Figure 1 illustrates a regimen of a tapering dose of Prednisone. RESUME could detect the “tapering” property of the Prednisone drug course abstraction. It would be extremely difficult and tedious to detect this temporal property using our SQL-based approach.

Haimowitz and Kohane developed a similar temporal abstraction system, using a frame-based constraint system based on trend templates.[14, 15] Like the RESUME system, describing, detecting, and creating the required interval abstractions and detecting the “tapering” state of the Prednisone abstraction would be easily done using trend templates.

An examination of Table 1 reveals that for pharmacy-related events, our underlying temporal data model is interval-based. Das has developed and implemented an extended relational database system, called CHRONUS, which directly supports interval-based temporal models.[5, 8] Das motivates the need for CHRONUS by highlighting special considerations for manipulating temporal attributes in the standard relational data model. Our experience confirms Das’ assertions – the

application programmer assumes significant responsibility for ensuring that temporal operations on temporal attributes are consistent. In addition, the complexity of expressing even simple temporal comparisons in standard SQL makes this application task even more difficult.

One gratifying outcome of our work has been the demand for our data from other interested users within the Department of Pharmacy. The unified abstractions of drug intervals and courses has created a method of visualizing and querying pharmacy-related data that has not previously been available. When coupled to commercial end-user querying tools, the user community has found this information to be able to answer questions not previously amenable to querying in the source systems.[12]

CONCLUSIONS

Despite the existence of temporal information in three clinical information systems, differences in temporal representations made it difficult to exploit these data. We have successfully mapped the original representations into a unified global representation and into a set of specific temporal abstractions. This approach, while requiring significant system-specific processing, has resulted in a new information resource for use by expert systems technology and ad-hoc end-user queries.

ACKNOWLEDGMENTS

Supported in part by National Cancer Institute Grant 5-U01-CA60245, and National Library of Medicine Grants 5-R29-LM05387 and U01-LM05845.

References

1. Weyl S, Fries J, Wiederhold G, Germano F. A Modular Self-describing Databank System. *Comp Biomed Res* 1975;8:279-293.
2. Tansel AU, Clifford J, Gadia S, Jajodia S, Segev A, Snodgrass R. *Temporal Databases: Theory, Design, and Implementation*. Redwood City, CA: Benjamin Cummings Publishing Company, 1993.
3. Pissinou N, Snodgrass RT, Elmasri R, et al. Towards an Infrastructure for Temporal Databases: Report of an Invitational ARPA/NSF Workshop. *SIGMOD Record* 1994;23(1):35-51.
4. Kahn MG. Modeling Time in Medical Decision-Support Programs. *Medical Decision Making* 1991;11:249-264.
5. Das AK, Tu SW, Purcell GP, Musen MA. An extended SQL for Temporal Data Management in Clinical Decision-Support Systems. In: Frisse ME, ed. *Proceedings Symposium on Computer Applications in Medical Care*. New York NY: 1992:128-132.
6. Jensen CS, Clifford J, Elmasri R, et al. A Consensus Glossary of Temporal Database Concepts. *SIGMOD Record* 1994;23(1):52-64.
7. Ahn I, Snodgrass R. Performance Evaluation of a Temporal Database Management System. In: Zaniolo C, ed. *Proceedings, ACM/SIGMOD International Conference on Management of Data*. New York NY: 1986:96-107.
8. Das A, Musen M. A Temporal Query System for Protocol-Directed Decision Support. *Meth Inform Med* 1994;33:358-370.
9. Abrams C, Steib S, Reichley R, Brennan T, Bailey T, Kahn MG. An Expert System for Monitoring Renally Excreted Drugs. To appear, *Proceedings, Symposium on Computer Applications in Medical Care* 1995.
10. Marrs KA, Steib SA, Abrams CA, Kahn MG. Unifying Heterogeneous Distributed Clinical Data in a Relational Database. In: Safran C, ed. *Proceedings Symposium on Computer Applications in Medical Care*. New York, NY: 1993:644-648.
11. Marrs KA, Kahn MG. Extending a Clinical Repository to Include Multiple Sites. To appear, *Proceedings, Symposium on Computer Applications in Medical Care* 1995.
12. Steib SA, Reichley RM, McMullin ST, et al. Supporting Ad-Hoc Queries in an Integrated Clinical Database. To appear, *Proceedings, Symposium on Computer Applications in Medical Care* 1995.
13. Shahar Y, Musen MA. RESUME : A Temporal-Abstraction System for Patient Monitoring. *Comp Biomed Res* 1993;26:255-273.
14. Haimowitz IJ, Kohane IS. Automated Trend Detection with Multiple Temporal Hypotheses. In: Bajcsy R, ed. *Proceedings International Joint Conference on Artificial Intelligence*. Boston MA: 1993:146-151.
15. Haimowitz IJ. Intelligent Diagnostic Monitoring Using Trend Templates. In: Ozbolt J, ed. *Eighteenth Annual Symposium on Computer Applications in Medical Care*. Washington, DC: Hanley-Belfus, 1994:702-708.